DII COE for Real Time: Becoming Reality

Lt. Col. Lucie M.J. Robillard
U.S. Air Force

Dr. H. Rebecca Callison and Marilynn B. Goo
The Boeing Company

John Maurer
MITRE Corporation

The Defense Information Infrastructure (DII) Common Operating Environment (COE) provides an environment in which common reusable infrastructure and applications across information systems help achieve goals for interoperability. The Department of Defense has been working for the past three years toward realization of a vision for extending these reuse and commonality initiatives to improve the effectiveness of systems performing real-time missions. This article describes the products, processes, tools, and techniques that have been developed to meet the needs of the integrator of DII COE-compliant real-time systems.

The following opening statement from a September 1999 CrossTalk article began a presentation on the motivation for and objectives of an effort to extend DII COE to real-time systems:

> "The Defense Information Infrastructure (DII) Common Operating Environment (COE) originated with a simple observation about command and control systems: Certain functions (mapping, track management, communication interfaces, etc.) are so fundamental that they are required for virtually every command and control system. Yet these functions are built over and over again in incompatible ways even when the requirements are the same or vary only slightly between systems. If these common functions could be extracted, implemented as a set of extensible building blocks, and made readily available to system designers, development schedules could be accelerated and substantial savings could be achieved through software reuse. Moreover, interoperability would be significantly improved if common software were used across systems for common functions [1]."

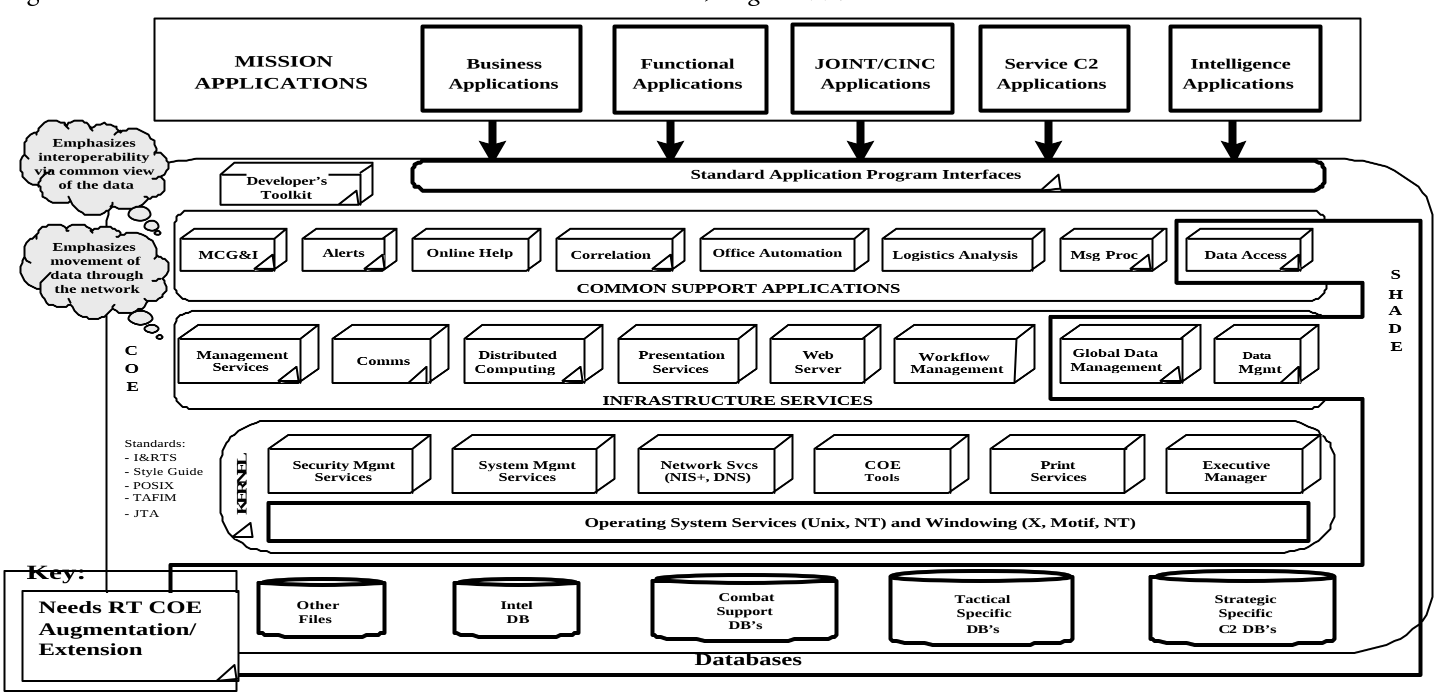
In this article we report on the status of those activities. Real-time processing is defined as a computation whose correctness depends on being logically correct and complete by a designated time. In a real-time system, the time that a process completes its computation and delivers its results is as important to correctness as, for example, the precision or accuracy of the answer. What is important is not *only* how fast the system responds but that it responds at the appropriate time. A protocol for synchronizing clocks across a communication network (distributed time service) is required to be accurate and timely, not just fast. The next generation of real-time systems will be so complex that more technical sophistication, not raw speed, will be the critical factor.

Extending DII COE for Real Time

In 1996 at the U.S. Air Force Electronic Systems Center, Hanscom AFB, Mass., all command and control programs were asked to develop a set of requirements for real-time extensions to existing DII COE capabilities. In the spring of 1997, representatives from the Air Force, Army, and Navy met to discuss the high correlation of real-time requirements across the services. In July 1997 these three services along with the Marine Corps jointly petitioned the Defense Information Systems Agency (DISA) to charter a DII COE Real-Time Technical Working Group (RT, TWG), with an aim of developing a set of common requirements and recommendations for potential products to provide real-time capabilities to the DII COE. DISA approved the services' request, and the Real-Time TWG began meeting in August 1997.

Initial studies conducted at Electronic

Figure 1: *Architectural Vision For DII COE Real-Time Extensions, August 1997*



Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **OCT 2001**	2. REPORT TYPE	3. DATES COVERED **00-00-2001 to 00-00-2001**
4. TITLE AND SUBTITLE **DII COE for Real Time: Becoming Reality**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **ESC/AWPD,3 Eglin Street,Hanscom AFB,MA,01730**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES **CROSSTALK The Journal of Defense Software Engineering, October 2001**		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

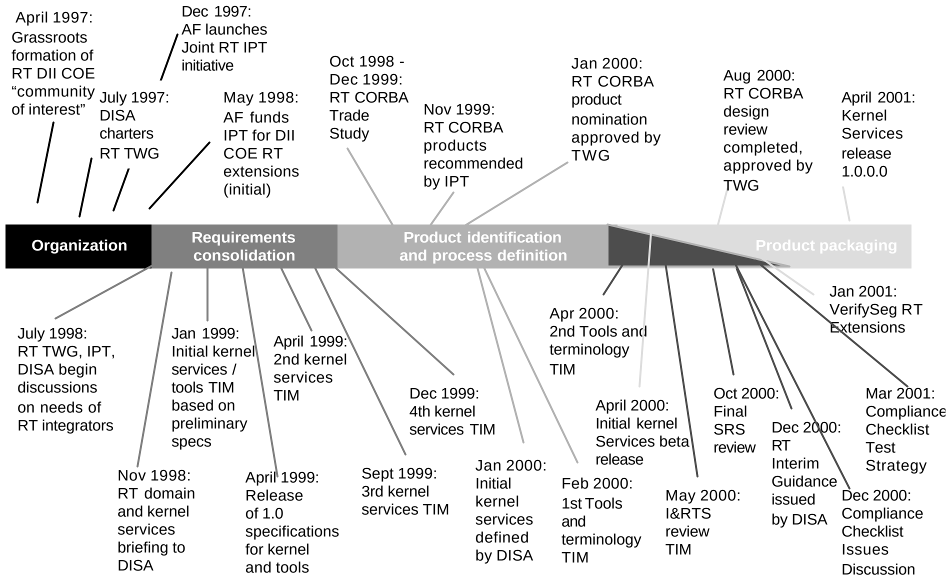


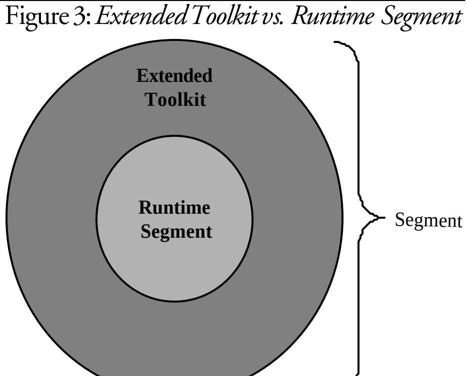
Figure 2: *Evolution Of DII COE Real-Time Extensions*

Systems Center highlighted numerous, relevant characteristics of real-time systems and suggested that a non-orchestrated approach to assembling real-time components would not be effective. In late 1997, the Air Force designated the Airborne Warning and Control System (AWACS) Program Office as Executive Agent for DII COE Real-Time Extensions. The DII COE Joint Real-Time Integrated Product Team (Joint RT IPT) embodies that executive authority. Because their missions are so closely related, the RT TWG and IPT have worked in continuous coordination, conducting joint meetings and sharing data. Both the TWG and IPT enjoy the active support and participation of Army, Air Force, Navy, and intelligence community representatives, all focused on the vision that is captured in Figure 1 (see page 19). The products, processes, tools, and techniques described in this article are the result of the activities of these two groups working in collaboration with DISA.

Real-Time Extensions

As depicted in the timeline of Figure 2, the anticipated release of a DII COE with real-time extensions represents the culmination of nearly four years of effort on the part of DISA, the RT TWG, and the Joint RT IPT.

Figure 3: *Extended Toolkit vs. Runtime Segment*



Initially, the activity of the RT TWG focused on collecting requirements from the RT community and consolidating them in software requirements specifications. As an understanding of these requirements matured, the TWG engaged the DISA DII COE engineering office in translating the requirements into beta implementations of a DII COE kernel and segment development tools with real-time extensions. The RT TWG also worked with DISA to modify its processes to support inclusion of real-time components in the DII COE. In parallel, the RT IPT focused on finding and packaging components required by the real-time community to provide capability above the DII COE kernel.

The specifications developed by the RT TWG and submitted to DISA became the basis for the real-time kernel and real-time extensions to the segment development tools. In 2001, DISA's DII COE-wide kernel TWG assumed control of the RT TWG specifications for real-time kernel services and real-time extensions to the segment development tools. The RT TWG also specified requirements for tools to aid the real-time systems integrator. DISA's Toolkit TWG assumed responsibility for the real-time extensions to the integration tools specification.

Evolution of DII COE Real-Time Extensions

Segmentation Concepts

Segmentation concepts encompass both the packaging constraints and the tools provided by DISA to ensure compatibility and peaceful coexistence among applications installed in a runtime (*mission*) environment. DISA's original segmentation concepts focused on executable binary applications and dynamically linkable libraries, i.e., the component formats that are used directly in the runtime environment of an operational system. This concept has been extended to support distribution of statistically linkable object libraries that may be combined with other components to produce tailored executable application images for the DII COE runtime environment.

Extended Toolkit Segments vs. Runtime Segments

Relating real-time requirements to the concept of segmentation resulted in a modified definition of segment, shown in Figure 3. *Segment* is defined by DISA in *Interim Guidance for DII COE Realtime Extensions* [2] as a "collection of one or more software and/or data units most conveniently managed as a unit of functionality." A runtime segment is a "segment that has been stripped of extraneous files and directories that are not required for a runtime target system."

A runtime segment corresponds roughly to the classic definition of a DII COE segment: software, data, and configuration information that will become part of and are used in the DII COE runtime environment. An extended toolkit is a "segment that contains documentation, shared libraries or those able to be linked, data, and other items required for use in an integration, development, and/or runtime environment."

The extended toolkit includes the runtime segment plus additional information needed to produce a runtime executable application. The extended toolkit enables DII COE software to be efficiently integrated into complex weapon systems in an integration environment prior to installing the software for operational use. This allows engineers to optimize DII COE compliant software for peak performance, a step that the existing DII COE did not previously support.

Extended toolkits and runtime segments are both *segments*. In general, a runtime segment is a proper subset of an extended toolkit segment. The classic DII COE runtime segment is delivered to a system integrator in the format accepted by the DII COE Installer tool. An extended toolkit is delivered in *tar* format and is loaded into the integrator's development

and/or integration environment using simple operating system utilities. It is important to note that the definitions of extended toolkit segment and runtime segment apply across the DII COE and are not unique to real time.

Figure 4 provides the directory structure of a generic extended toolkit segment. In this figure, a dotted line box contains the items that are also applicable to a runtime segment. The parts of the extended toolkit that are used in the integration environment are marked with diagonal lines. Similarly, the light colored boxes show directories that are used in the development environment and the darker boxes show directories that are used in the runtime environment.

As an aid in the configuration of full systems from DII COE components, the dependencies of DII COE real-time segments on other DII COE segments, kernel services, and real-time operating system (RTOS) Portable Operating System Interface (POSIX)[1] units of functionality are documented in the segment descriptors provided by the segment supplier. A designer of a real-time system can then match system requirements to choices of DII COE applications, infrastructure, kernel services, and RTOS.

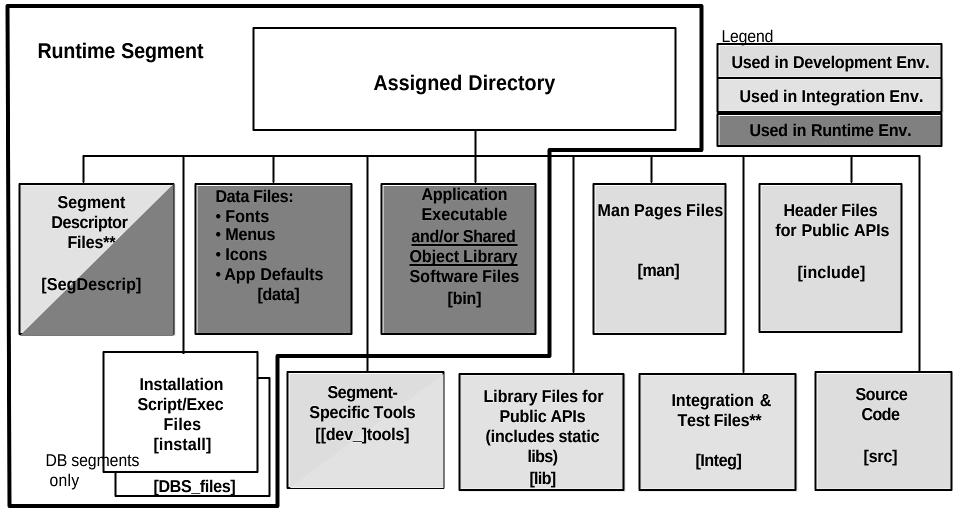


Figure 4: *Extended Toolkit Segment Directory Structure*

Additions Motivated by Real Time

A more in depth look at the segment contents uncovers additions, in Figure 5, that were motivated by the inclusion of real-time segments. Extensions to the SegInfo file in the SegDescrip directory describe additional hardware dependencies (e.g., shared memory and special hardware devices) and software dependencies (e.g., POSIX Units of Functionality) that may limit the execution environments in which segments can run. Beyond the addition of specific keywords to SegInfo, the attribute *<:restricted>*, when appended to *$CPU* or *$OS* identifiers, tells the integrator that the segment may place unusual restrictions on the operating environment. These restrictions must be fully documented in the IntgNotes file that is included in the Integ directory.

The IntgNotes file has been extended significantly to include items of key interest to integrators of real-time systems. Enhancements have two principal forms: freestanding additions and additions that explain or elaborate conditions noted in the extended SegInfo file. Examples of the former include IntgNotes entries under which real-time scheduling policies and restrictions, scheduling frequencies, jitter tolerance, CPU utilization, and other aspects of real-time behavior may be described. Examples of the additions which explain SegInfo entries include 1) entries under which <:restricted> notations applied to $CPU or $OS keywords are fully described and 2) entries in which the rationale for CPU, memory, and disk resource requirements are documented. The IntgNotes file remains a free-format text file into which the segment developer may insert any information deemed of potential interest to the system integrator using the segment.

As with the basic segmentation concepts, it is important to remember that none of the SegInfo or IntgNotes extensions are real-time specific. Some of this information is *required* in conjunction with real-time segments. For other segments, use of the extensions is optional.

Real-Time Products

Several products will be available for use in 2001 that provide real-time performance and are part of DII COE. Figure 6 (see page 22) shows how each of the products relates to Figure 1 (see page 19).

DII COE Real-Time Kernel and Platforms

The DII COE kernel provides the basic interfaces and functions to be used by standards-based infrastructure components and DII COE-compliant applications to achieve portability between systems. The DII COE configurable real-time kernel[2] extends basic DII COE concepts in two ways. First, the RT Kernel is hosted only on operating systems that provide real-time scheduling capabilities, reasonably predictable operating system performance, and the services required for timely execution of real-time tasks and processes. RT Kernel services are selectable, rather than mandatory.

Second, since real-time applications often need a very efficient operating system with small memory footprint for performance reasons, the design philosophy of the DII COE RT Kernel allows a system integrator to tailor the RTOS itself to meet system needs.[3] The RT Kernel is configurable and the integrator of a DII

Figure 5: *Process and Information Extensions*

DII COE Segmentation with Extended Toolkits

- **Guidance Documentation**
  - ✦ Augmented I & RTS
  - ✦ Interim Guidance for RT
- **Example Segments**
  - ✶ Add example RT segments
- **Segment Descriptor Files**
  - ✦ Enhanced SegInfo
  - ✦ Enhanced IntgNotes
- **Segmentation Tools**
  - ✦ Extended VerifySeg for new descriptions
  - ✶ Extended VerifySeg for RT attribute checks
- **Dependencies defined in SegInfo descriptors**
  - ✦ Segments/Toolkits
  - ✶ Real Time Operating System
    - New HW Platforms
    - Disk Space
    - Memory
  - ✦ POSIX Units of Functionality
  - ✦ Custom Hardware Devices
  - ✦ Shared Memory
- **Installation Process**
  - ✦ Installation in Integration Environment (Not Runtime)
- **Segment Repository**
  - ✶ Port some DII COE Segments to RT COE

✶=extension used uniquely by RT segments
✦=extension motivated by real-time requirements

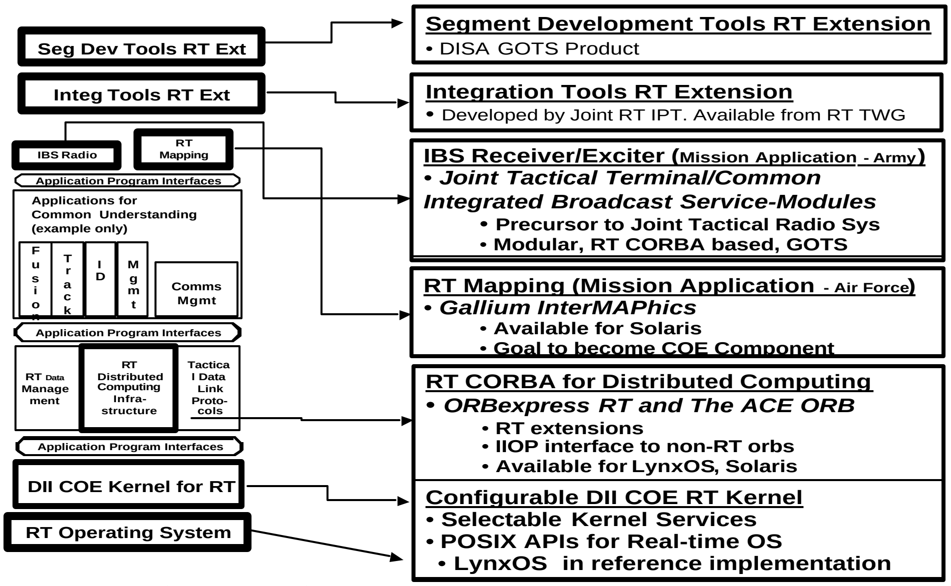


Figure 6: *Products Projected as DII COE Compliant Real-Time Segments*

COE-compliant system tailors the RT Kernel by selecting only those services required for the specific computing configurations of the target system. POSIX Application Program Interface (APIs) for operating system services, including APIs for threads and real-time extensions specified in [3], form part of the RT Kernel API. Each RTOS being considered for use in the DII COE will be assessed for its ability to provide key functional units associated with real-time profiles in the POSIX 1003.13 standard [4]. LynxOS[4] Version 3.1.0a running on PowerPC was selected as the reference (i.e. first) implementation. Sun Solaris[5] Version 8 is the second real-time capable operating system on which the RT Kernel is hosted.

As noted earlier, the RT Kernel has two parts: 1) a RTOS with POSIX application program interfaces and 2) selectable DII COE kernel services for real time. The RT Kernel services are provided by DISA. DII COE for real time includes X, Motif, and Domain Name Server, all of which are commercial off-the-shelf products, plus government off-the-shelf services for system startup and shutdown, setting system time, and starting and stopping DII COE processes. Requirements for the RT Kernel services are documented in [5].

CORBA Infrastructure for Real Time

Common Object Request Broker Architecture (CORBA) is an international standard [6] for distributed computing that is governed by the Object Management Group. The CORBA standard provides for flexible interconnection of objects in a client-server model for distributed computing. Four key objectives of CORBA are support for location independence, operating system independence, hardware independence, and language independence in the design of software components.

Additions to the CORBA standard to enable real-time computing with end-to-end predictability are documented in the CORBA specification revision 2.4 [6], which was formalized by the Object Management Group in October 2000. These additions allow for the association of real-time priorities with tasks and requests, the passing of priority information between communicating components, and the capability to express and monitor timing constraints for requests. These additions also define a scheduling service that provides a consistent real-time scheduling model across a CORBA-based system.

In 1999, a real-time CORBA trade study was performed by the Joint RT IPT to assess products being considered for use by the real-time community. The goal of the study was to determine whether or not any or all of these products would be suitable for use in the real-time extensions of the DII COE. The study included five assessments. The technical assessments addressed three areas: standards compliance, basic performance, and interoperability with other object request brokers (ORBs). The other assessments included a user survey of product usability and a cursory examination of the business viability of each vendor and product. At the time of the assessments, the real-time additions to the OMG CORBA standard were still in development and none of the vendors had implemented them. The study can be obtained from <www.hanscom.af.mil/foia/misclist.asp?contractid=3&descrip tion=Other+Documents>.

Based on recommendations from the real-time CORBA trade study, the RT TWG nominated two real-time ORBs for inclusion in the DII COE infrastructure.[6] The real-time ORBs are ORB*express* RT[6], a product of Objective Interface Systems, Herndon, Va., and The ACE ORB (TAO)[7], an open source product of Washington University, St. Louis, Mo., that is commercially supported by Object Computing, Inc., also in St. Louis. ORB*express* RT supports both Ada and C++ and includes extensions for real-time performance. TAO supports C++ and also includes extensions for real-time performance. Both are cognizant of real-time request priorities and provide the capability to associate deadlines with requests. Vendor packaging of these products as extended toolkit segments began in 2000 and was completed in mid-2001.

Real-Time Mission Applications: A real-time mapping product is available as a DII COE-compliant AF mission application during 2001: InterMAPhics[8], a product of Gallium Software, Inc., Nepean, Ontario, Canada. This product may someday be in the DII COE Common Support Applications layer. The decision to proceed in this direction depends on the capabilities of the product selected by the National Imagery and Mapping Agency under the Commercial Joint Mapping Toolkit procurement. In the meantime, the real-time community can use this high performance product as a real time alternative to the Joint Mapping Visualization part of Joint Mapping Toolkit.

The Army is developing a software programmable exciter/receiver called the Joint Tactical Terminal/Common Integrated Broadcast Service-Modules (JTT/CIBS-M) under contract with Raytheon, St. Petersburg, Fla. This software is being packaged as a DII COE-compliant Army mission application during 2001. JTT/CIB-M supports a variety of intelligence broadcast protocols such as TDDS, TADIX-B, TIBS and TRIXS.

DII COE Real-Time Tools

Based on the RT TWG specification for real-time extensions to segment development tools, DISA has modified its VerifySeg tool to check the additions to the SegInfo file segment descriptor information described earlier in Figure 5 (see page 21). This version of VerifySeg is used by developers of runtime segments as well as by developers of extended toolkit segments. The output from VerifySeg becomes part of the segment. VerifySeg and the requirements for its use are

described in DISA's *Integration and Runtime Specification (I&RTS)* [7].

In the fall off 2000, the Joint RT IPT began developing the real-time integration tools to prototype the integration process associated with developing a DII COE compliant real-time system. These tools are intended for use by software system integrators in the integration environment. Since the COE installer tool cannot be used to install runtime segments in many real-time runtime (target) environments, some automated assistance is needed for ensuring a proper configuration of DII COE segments. The real-time integration tools provide this assistance by analyzing the intended segment configuration for inter-segment dependencies and conflicts. For embedded systems, they also assist the integrator in configuring (scaling down) the operating system (OS) to include only those OS functions needed to support the target application software load.

The integrator supplies a list of the capabilities to be configured on a target system by selecting from a list of available segments. The primary output of the real-time integration tools is a real-time configuration (RTConfig) file that lists all segments, including kernel services that must be loaded on the target platform in order for the selected segments to run. Using information contained in segment SegDescrip directories, the real-time integration tools expand the list of selected functions based on the dependencies of runtime and extended toolkit segments on other segments. For example, if segment A depends on segment B, and segment B depends on segment K, then the RTConfig file will include segments A, B, and K. The real-time integration tools also produce a list of POSIX capabilities (e.g., POSIX units of functionality) required by these segments in the OS configuration. If any of the segments have conflicts noted in their SegInfo files, this information is included in the RTConfig file.

The real-time integration tools were completed in June 2001. Acceptance of these tools by DISA depends on the customer demand for the tool and the value added seen by DISA. In the meantime, the RT TWG makes the tools available to interested users upon request via the RT TWG Web page <www.dii-af.hanscom.af.mil/infrastructure/COE/TWG/COE/TWG/rtcoe/NewTWG/index.htm>.

DISA Process

Real-Time Interim Guidance

The new capabilities that are available with the DII COE real-time extensions have required that the rules governing the development of DII COE segments be modified. These new capabilities include extensions to exploit the features of a real-time platform, the configurable kernel, and development of extended toolkit segments intended for delivery to an integration environment. The existing rules are documented in DISA's *Integration and Runtime Specification (I&RTS)* [7]. The modifications to these rules have been published in the *Interim Guidance for Defense Information Infrastructure Common Operating Environment (COE) Realtime Extension* [2], which will be refined through initial practical experience and incorporated into Version 5.0 of the I&RTS.

The interim guidance document provides detailed information that a segment supplier needs to develop segments intended to run on the RT Kernel. In addition, the interim guidance document provides a preview of how the rules may be applied in the next major release, DII COE Version 5.0. The interim guidance document provides a discussion of definitions and concepts, as well as an updated version of the compliance checklist criteria (aka, Appendix B) as it applies to the real-time platform. It is available in the technical baseline section of the RT TWG Web site <www.dii-af.hanscom.af.mil/infrastructure/COE/TWG/COE/TWG/rtcoe/NewTWG/Baseline/DIICOERTEInterimGuidance.PDF>.

Toolkit Compliance Evaluation

Compliance evaluation for an extended toolkit requires a different perspective than for a classic DII COE runtime segment. The latter is installed directly in the runtime (target) environment, whereas the former is not – it is loaded first into an integration environment. In both cases (runtime segments and extended toolkit segments), compliance evaluation is intended to ensure correct *runtime* behavior. For runtime segments, the evaluation can be performed in the target (runtime) environment. However, in the case of extended toolkits, the compliance evaluation must, in general, be performed in a development or integration environment with an eye toward how a *runtime* segment *built from the extended toolkit* will behave in the *runtime* environment.

When extended toolkits contain static libraries only, there is no clearly identifiable *runtime segment* subset of the toolkit that will *ever* be installed directly into the runtime environment. Rather the integrator will *always* link the libraries of these toolkits with other applications and toolkits to produce the target system executables. However, in order to ensure that the integrator's customized executables satisfy the constraints of the DII COE environment, the behavior of the delivered libraries, as contributors to that behavior, must be scrutinized. For example, a number of compliance criteria address constraints on the creation of files outside the DII COE directory structure. If an application links to a library that creates files in a non-compliant location, the application executable can never be compliant. Since each compliance criterion that affects the application executable must be flowed down to the library in the extended toolkit as well, the problem becomes one of defining the details of how to evaluate compliance for toolkits.

After the release of [2], the RT TWG was asked to determine for each item in the interim guidance whether or not that Appendix B compliance checklist item applies to an extended toolkit segment. The RT TWG was also asked to determine the test strategy for each applicable item. This action was completed in early 2001. The resulting document, which is available at <www.dii-af.hanscom.af.mil/infrastructure/COE/TWG/COE/TWG/rtcoe/NewTWG/baseline.htm>, is the RT TWG's recommendation for ensuring correct runtime behavior while performing compliance evaluation of extended toolkits in an integration environment.

A Turnkey Segmentation Process

Segmenting software for the DII COE is a recurring task where attention to detail can prevent rework. The RT IPT has been working with various services, agencies, DISA, and segment suppliers to define the steps necessary to enable real-time products to become DII COE compliant. Lessons learned as a result of this effort are being recorded in the form of tutorials and templates. Table 1 is a partial list of prod-

Tutorials	Templates
Segmentation Overview	DISA Design Review Questions
Segmentation	Segmentation Plan
Prefix & Segment Registration	IntgNotes

Table 1: *Turnkey Segmentation Products*

ucts available from the RT TWG for use by those who need to prepare extended toolkit segments for the DII COE.

DII COE Real-Time's Future

In 2001, several changes have occurred regarding the future of this effort. DISA has changed the name of the RT TWG to the Real-Time Advisory Group (RTAG). There will be slight modifications to the charter for this group. In addition, the Joint RT IPT has completed its tasks and transitioned its remaining responsibilities to the RTAG (as of 30 June 2001). So the RTAG will be the main point of contact for the real-time community regarding DII COE capabilities and requirements.

The RTAG remains under the umbrella responsibility of the DISA DII COE Chief Engineer Office. The chairman of the RTAG remains on the DII-Air Force Office staff at Hanscom AFB, Mass., and continues to serve all services and agencies.

Based on the work of the Joint RT IPT and RTAG in the last four years, it is envisioned that the future products needed by the real-time community in the DII COE include real-time data access, Link 16, Joint Variable Message Format (JVMF) Parser, Data Correlator, and Alerts. It is the responsibility of each system program office and their sponsoring service or agency to work with the RTAG to sponsor appropriate government or commercial off-the-shelf products into the DII COE. This responsibility will often include performing the work needed to create a segmented product according to DISA's I&RTS [7]. The turnkey segmentation package created by the Joint RT IPT is a valuable resource in accomplishing this task.

Conclusion

In four years, a small contracted engineering effort supplemented by a strong team of engineers working on donated funding have accomplished key tasks that make it possible for real-time weapon systems to pursue increased command and control interoperability via DII COE compliance. The basic vision has been accomplished.

The Joint RT IPT provided solid, reliable technical leadership and support for a fast-paced, dynamic program fraught with technical challenges. The real-time community can now directly utilize the lessons learned from this effort to bring more software products into the DII COE. The foundation has been set for more work to be done in support of the real-time community by working with the RTAG. As the real-time effort is normalized into the mainstream of service acquisition agencies, the future enhancements of DII COE for real time lie in the hands the real-time weapon system builders and their customers, and the services and agencies with real-time command and control missions.◆

References

1. Robillard, Lt. Col. Lucie M.J., Dr. H. Rebecca Callison and John Maurer. "Extending the DII COE for Real-Time," CrossTalk, Sept. 1999.
2. Interim Guidance for Defense Information Infrastructure Common Operating Environment (COE) Realtime Extension, Version 1.0, Defense Information Systems Agency, Dec. 2000.
3. "Portable Operating System Interface (POSIX) Part 1 – System Application Program Interface (API) [C Language]," *Information Technology*, ISO/IEC 9945-1:1996 (E) ANSI/IEEE Std. 1003.1.
4. Portable Operating Systems Interface (POSIX) 1003.13-1998, IEEE Standard for Information Technology Standardized Application Environment Profile (AEP)–POSIX Realtime Application Support.
5. Software Requirements Specification for Kernel Services for the Real-Time Defense Information Infrastructure Common Operating Environment (RT DII COE), Revision 1.5, DII COE RT TWG, 30 Jan. 2001, <www.dii-af.hanscom.af.mil/infrastructure/COE/TWG/COE/TWG/rtcoe/NewTWG/baseline.htm>.
6. The Common Object Request Broker: Architecture and Specification, Revision 2.4, Object Management Group, Inc., Oct. 2000, <http://cgi.omg.org/cgiin/doc?formal/00-10-01>.
7. Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification, Version 4.1, DISA, DISA Joint Interoperability and Engineering Organization, Reston, Va., Aug. 2000.

Notes

1. POSIX is a registered trademark of The Institute of Electrical and Electronic Engineers, Inc. (IEEE).
2. In the rest of this paper, we will use the term "RT Kernel" as an abbreviation for "DII COE Configurable RT Kernel."
3. COTS tools provided by the OS vendor are used to configure the OS, not DII COE unique software. The degree to which a specific RTOS can be configured depends on the flexibility provided by the RTOS vendor.
4. LynxOS is a trademark of LynuxWorks, Inc. <http://www.lynuxworks.com>.
5. Sun and Solaris are trademarks of Sun Microsystems, Inc. <http://www.sun.com>.
6. ORBexpress RT is a trademark of Objective Interface Systems <http://www.ois.com>.
7. The ACE ORB is a trademark of Douglas Schmidt, Washington University, and the University of California, Irvine <http://www.cs.wustl.edu/~schmidt/TAO.html>.
8. InterMAPhics is a trademark of Gallium Software, Inc. <http://www.gallium.com>.

About the Authors

Lt. Col. Lucie M.J. Robillard is the Air Force executive agent for Real - Time DII COE at Hanscom AFB, Mass. She is the chair for the DII COE Joint Real-Time Integrated Product Team that has the charter to make real-time extension to DII COE a reality for all services. She is a Level III certified acquisition professional with Joint assignment experience. A majority of her assignments have dealt with software acquisition and engineering. She has a bachelor's degree in electrical engineering from the University of Vermont and a master's of science degree in systems management from University of Southern California. As of July 1, 2001, she has moved on to another assignment.

ESC/AWPD
3 Eglin Street
Hanscom AFB, MA 01730
Phone: (781) 377-2679
E-mail: lucie.robillard@hanscom.af.mil

Marilynn B. Goo is the program manager for the Boeing team supporting Lt. Col. Robillard and the DII COE Joint Real-Time Integrated Product Team. Goo has 26 years of experience in design and implementation of complex systems, principally in the area of defense systems. She has a bachelor's of science in mathematics from the University of Washington and an master's of business administration from Stanford University.

The Boeing Company
P.O. Box 3999, MS 82-84
Seattle, WA 98124
Phone: (253) 773-9867
Fax: (253) 657-2892
E-mail: marilynn.b.goo@boeing.com

H. Rebecca Callison, Ph.D., is the software architect for the U.S. Airborne Warning and Control System (AWACS) Block 40/45 Program. Previously, she was the lead for the Boeing team supporting Lt. Col. Robillard and the DII COE Joint Real-Time Integrated Product Team. Dr. Callison has 25 years of experience in the design and implementation of real-time systems, principally in the area of defense systems. She has a bachelor's of science degree from the University of South Carolina, a master's of science engineering from the University of Pennsylvania, and a doctorate from the University of Washington. Her research interests focus on concurrency control in real-time systems.

The Boeing Company
P. O. Box 3999, MS 81-75
Seattle, WA 98124
Phone: (253) 657-3952
Fax: (253) 657-4269
E-mail: rebecca.callison@boeing.com

John Maurer leads MITRE's Real-Time and Performance Engineering Section. Maurer is also the chairperson of the DII COE Real-Time Advisory Group. He has a bachelor's of science in mechanical engineering from Massachusetts Institute of Technology and 24 years experience implementing software-intensive Department of Defense systems. His work experience includes real-time system development for airborne surveillance systems and Army vehicle systems.

MITRE Corporation
202 Burlington Road
Bedford, MA 01730
Phone: (781) 271-2985
Fax: (781) 271-4686
E-mail: johnm@mitre.org

Coming Events

November 4-7
Amplifying Your Effectiveness Conference 2001
Phoenix, AZ
www.ayeconference.com

November 4-8
ASIS 2001 Annual Conference
Washington, D.C.
www.asis.org

November 6-8
TechNet Asia-Pacific 2001
Honolulu, HI
www.afcea.org

November 12-16
5th International Software and Internet Quality Week - Europe 2001
Brussels, Belgium
www.qualityweek.com

November 13-15
1st Annual CMMI Technology Conference and User Group
Denver, CO
djenks@ndia.org

January 27-31, 2002
2002 Western Multiconference
San Antonio, TX
www.scs.org

February 4-6, 2002
International Conference on COTS-Based Software Systems (ICCBSS)
Lake Buena Vista, FL
www.iccbss.org

February 25-27, 2002
15th Conference on Software Engineering Education and Training (CSEE & T)
Covington, KY
www.spsu.edu/oce/cseet2001

March 19-21, 2002
Federal Office Systems Exposition 2002
Washington D.C.
www.fose.com

April 28 - May 2, 2002
Software Technology Conference 2002 "Forging the Future of Defense Through Technology"
Salt Lake City, UT
www.stc-online.org